OULUN SEUDUN
AMMATTIKORKEAKOULU

Tero Kääntä

HTTP-PALVELINOHJELMIEN VERTAILU RASPBERRY PI:SSÄ

HTTP-PALVELINOHJELMIEN VERTAILU RASPBERRY PI:SSÄ

Tero Kääntä
Opinnäytetyö
Lukukausi Syksy 2013
Tietojenkäsittely
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu

Tietojenkäsittelyn koulutusohjelma

Tekijä: Tero Kääntä

Opinnäytetyön nimi: HTTP-palvelinohjelmien vertailu Raspberry Pi:ssä

Työn ohjaaja: Jukka Karlström

Työn valmistumislukukausi ja -vuosi: Syksy 2013 Sivumäärä: 41

Tämän opinnäytetyön tarkoituksena oli vertailla HTTP-palvelinohjelmien Apachen ja Nginxin suorityskykyä sekä selvittää miten niiden nopeutta voidaan optimoida. HTTP-palvelinohjelmat asennettiin luottokortin kokoiselle Raspberry Pi tietokoneelle, joka on edullinen ja vähän virtaa kuluttava vaihtoehto HTTP-palvelimeksi kotikäyttöön. Suorituskykyä testattiin Siege-ohjelmalla, joka kuormittaa HTTP-palvelinta virtuaalikäyttäjien avulla. Raportissa kerrotaan HTTP-palvelimista ja niihin liittyvistä ohjelmista. Opinnäytetyössä tehdyt asennukset ja optimoinnit ovat dokumentoitu raporttiin.

Raspberry Pi ja toinen tietokone, jolla kuormitustestit ajettiin, olivat samassa lähiverkossa. Opinnäytetyössä tehtiin kolme erilaista kuormitustestiä. Ensimmäinen kuormitustesti tehtiin täysin staattisille HTML-sivuille HTTP-palvelinohjelmien perusasennuksilla. Toisessa kuormitustestissä testattiin Wordpress-sivuja HTTP-palvelinohjelmien perusasennuksilla. Kolmannessa kuormitustestissä testattiin Wordpress-sivuja, kun HTTP-palvelinohjelmat oli optimoitu nopeammaksi. Nginx oli kaikissa testeissä nopeampi, mutta Apache hyötyi enemmän optimoinnista.

Opinnäytetyöstä selviää, miten paljon HTTP-palvelinohjelman suorituskyvyn optimointi vaikuttaa verkkosivujen latausnopeuteen, etenkin jos verkkosivut sisältävät PHP-koodia. Tämän opinnäytetyön avulla voidaan todeta Nginxin olevan nopeampi ja kevyempi HTTP-palvelinohjelma Raspberry Pi tietokoneella, jossa on vähän keskusmuistia ja hidas prosessori.

Asiasanat: HTTP-palvelin, Apache, Nginx, optimointi, Raspbperry Pi

ABSTRACT

Oulu University of Applied Sciences

Degree programme in Business Information Systems

Author: Tero Kääntä

Title of thesis: HTTP server comparison on Raspberry Pi

Supervisor: Jukka Karlström

Term and year when the thesis was submitted: Autumn 2013 Number of pages:41

The main purpose of this thesis was to compare HTTP servers Apache and Nginx performance and also find out how to optimize their speed. HTTP server applications were installed on a credit card sized Raspberry Pi computer which is cheap and has low power usage. These make it a good choice to use as HTTP server at home. The performance was tested with Siege program, which runs stress tests on HTTP server with the help of virtual users. This report tells about HTTP servers and programs related to them. All installations and optimizations are documented in this thesis.

Raspberry Pi and another computer which ran stress tests were connected to the same local area network. There were three different stress tests performed during the process of this thesis. The first stress test was made on completely static HTML sites with default installation of HTTP server applications. The second stress test was run on Wordpress sites with default installation of HTTP server applications. The third stress test was run on Wordpress sites when the HTTP server applications were optimized for better performance. Nginx performed better in every test but Apache benefited more from the optimization.

This thesis demonstrates how much HTTP server application optimizing affects loading speed of websites especially if they include PHP code. Based on this thesis, it can be stated that, compared with Apache, Nginx is faster and lighter HTTP server in Raspberry Pi computer which has low amount of RAM and slow processor.

Keywords: HTTP-server, Apache, Nginx, optimizing, Raspberry Pi

SISÄLLYS

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on selvittää, miten HTTP-palvelinohjelmistojen Apachen ja Nginxin suorituskykyä voidaan parantaa optimoinnilla. Ohjelmat asennetaan Raspberry Pi tietokoneelle, joka toimii HTTP-palvelimena tässä vertailussa. Raspberry Pi on edullinen luottokortin kokoinen tietokone, jossa on vähäinen virrankulutus. Opinnäytetyössä tehdään suorituskykytestejä kyseisille HTTP-palvelinohjelmille, joilla voidaan todeta optimoinnin hyödyt. Testien avulla voidaan myös vertailla kyseisten HTTP-palvelinohjelmiston nopeuseroja. Opinnäytetyössä kerrotaan myös Raspberry Pi tietokoneen taustasta ja ominaisuuksista.

Tässä opinnäytetyössä käytetään toista tietokonetta, jolla luodaan etäyhteys Raspberry Pi:lle. Laitteet ovat samassa lähiverkossa. Käyttöjärjestelmäksi Raspberry Pi:lle asennetaan Raspbian. Opinnäytetyössä kerrotaan mitä Raspberry Pi:lle kannattaa tehdä palvelinkäytössä, muun muassa muistin vapauttaminen näytönohjaimelta keskusmuistin käyttöön ja ainoastaan komentorivin käyttäminen.

Laitteiden ja ohjelmien asennukset ovat dokumentoitu ja sisältyvät opinnäytetyöhön. Opinnäytetyön teoriaosassa keskitytään HTTP-palvelimiin ja niiden ominaisuuksiin. Työssä kerrotaan miten asennukset ja optimoinnit on tehty. Lopuksi tehdään kuormitustestejä staattisilla ja dynaamisilla verkkosivuilla. Kuormitustestit tehdään Siege-ohjelmalla, joka on tehty verkkokehittäjien ja järjestelmienylläpitäjien työkaluksi testata ohjelmiaan ja järjestelmiään.

2 HTTP-PALVELIMET

HTTP-palvelimella voidaan tarkoittaa HTTP-palvelinohjelmaa tai tietokonetta, jossa HTTP-palvelinohjelma on käytössä. Tietokone muuttuu HTTP-palvelimeksi, kun siihen asennetaan HTTP-palvelinohjelma ja toimiakseen se tarvitsee verkkoyhteyden. HTTP-palvelin varastoi ja jakaa tiedostoja Internetissä (Webopedia 2013, hakupäivä 13.11.2013.)

HTTP-palvelin käyttää HTTP-protokollaa, jonka avulla se voi olla yhteydessä käyttäjien verkkoselaimien kanssa. HTTP-protokollan käyttö edellyttää yhteyden muodostamista asiakkaan ja palvelimen välille käyttäen TCP/IP verkkoprotokollia. HTTP-palvelimella on oma IP-osoite, jolla se on yhteydessä verkkoon. Yleensä kuitenkin HTTP-palvelin käyttää domain nimeä esim. www.example.com. Koska tietokoneet ymmärtävät vain numeroita, tarvitaan DNS-järjestelmä muuntamaan kirjoitetut domain nimet IP-osoitteiksi ja päin vastoin. Tietokoneen täytyy ensin hakea domain nimeä vastaava IP-osoite DNS järjestelmästä, ennen kuin TCP/IP yhteys voidaan luoda asiakkaan ja palvelimen välille (Wisegeek 2013, hakupäivä 5.10.2013.)

Kolme suosituinta HTTP-palvelinohjelmaa ovat The Apache Software Foundationin HTTP-server, Microsoftin ISS (Internet Information Server) ja Nginxin HTTP-server. Netcraftin 2013 tekemässä tutkimuksessa Apachea käyttää 44,89% kaikista maailman HTTP-palvelimista. Microsoftin ISS:n osuus oli 23.10% ja Nginxin 16.05%. Apache on kuitenkin menettänyt käyttäjiään Nginxille ja Microsoftin ISS:lle viimeaikoina (Netcraft 2013, hakupäivä 4.10.2013.)

HTTP-palvelimet ovat päällä ja yhteydessä Internetiin jatkuvasti. Suuret verkkosivustot käyttävät kuormantasausta, jossa useampi HTTP-palvelin kytketään yhteen. Kuormantasaus jakaa työkuorman palvelimien kesken ja tällä tavoin se takaa suorituskykyä ja luotettavuutta (Osoite 2013, hakupäivä 7.12.2013.)

2.1 HTTP

HTTP-palvelimet ja verkkoselaimet käyttävät tiedonsiirtoon hypertekstin siirto protokollaa (HyperText Transfer Protocol). HTTP on sovellustason protokolla. HTTP

toimii pyyntö-vastaus periaatteella asiakas-palvelin mallissa. verkkoselain on asiakas, joka lähettää pyyntöviestejä HTTP-palvelimelle, jossa kerrotaan mitä tietoja ollaan hakemassa. HTTP-palvelin lähettää vastauksen, joka sisältää tiedot (Goldstein, N. 2012.)

HTTP:ta käytetään siirtämään HTML-tiedostoja, kuvatiedostoja ja muita resursseja. Resurssi määritellään tietona, joka voidaan paikantaa URL:na (Uniform Resource Locator). Esimerkiksi teronsivut.com/web/image.jpg, jossa image.jpg on resurssi ja koko lause URL. Yleisin resurssityyppi on tiedosto (Goldstein, N. 2012.)

HTTPS (Hypertext Transfer Protocol Secure) on samanlainen kuin HTTP, mutta se käyttää SSL/TLS salausta. Tiedot lähetetään salattuja kanavia pitkin. Transport Layer Security (TLS) ja sen edeltäjä Secure Sockets Layers (SSL) ovat tietoturvaprotokollia, jotka turvaavat kommunikoimisen Internetin yli (Goldstein, N. 2012.)

2.2 Yleisiä HTTP-palvelimen asetuksia

Tässä kappaleessa kerrotaan yleisistä HTTP-palvelimen asetuksista, jotka sijaitsevat HTTP-palvelimen asetustiedostoissa. Seuraavat asetukset on hyvä olla oikein konfiguroitu.

2.2.1 Suurin samanaikainen käyttäjämäärä

HTTP-palvelinohjelmissa on mahdollista määritellä samanaikaisten käyttäjien määrä ja optimoida se sopivaksi. Asetusta ei kannata määrittää suurimmaksi mahdolliseksi mitä järjestelmä jaksaa pyörittää, koska asiakkaita palvellaan hitaammin ja ei välttämättä päästä suurempiin käyttäjämääriin. Suotavampaa olisi keskittyä pyyntöjen nopeaan palvelemiseen, kuin suurimpaan samanaikaisiin käyttäjiin (Smith, P. 2012.)

2.2.2 Keep-Alive

HTTP 1.0:ssa yleinen asetus on, että yhteys suljetaan asiakkaan ja web-palvelimen välillä, kun haettu resurssi on saapunut. Tässä on kuvattu prosessi kun asiakas hakee monta resurssia:

1. Asiakas avaa TCP yhteyden palvelimelle.

2. Asiakas lähettää pyynnön resurssista.
3. Palvelin lähettää resurssin takaisin ja sulkee yhteyden.
4. Asiakas avaa uuden TCP yhteyden.
5. Asiakas lähettää pyynnön toisesta resurssista.

Tämä on suorituskykyä haaskaava tapa, mikäli haettavia resursseja on enemmän kuin yksi. TCP yhteyksien uudelleen luonti yksittäisten resurssien välillä lisää viivettä sekä prosessorin ja keskusmuistin käyttöä palvelimella ja asiakkaalla. Ongelma tulee paremmin esiin, kun haettavat tiedostot ovat pieniä ja niitä on paljon.

Ongelman ratkaisuun on kehitetty HTTP/1.1 versio, jossa on Keep-Alive toiminto, joka ei sulje asiakkaan ja palvelimen välistä yhteyttä välittömästi yhden resurssin latauksen jälkeen. HTTP-palvelin ja asiakas säilyttävät yhteyden ja asiakas voi tehdä uusia pyyntöjä yhdistämättä joka kerta uudelleen. Yleinen asetus HTTP-palvelimissa on 5-10 sekuntia, jonka jälkeen palvelin katkaisee yhteyden jos uusia pyyntöjä ei tule. Asetusta on mahdollista muuttaa HTTP-palvelimen asetustiedostossa. (Smith, P. 2012.)

Keep-Alivea ei tarvita silloin, kun ladataan yksittäisiä HTML-sivuja, jotka ei sisällä muita resursseja. Esimerkiksi HTML-sivu, jossa on vain tekstiä. Asiakas lataa vain yhden resurssin ja sulkee yhteyden, jolloin se vapauttaa muistia palvelimelta. Tämänlaiset sivut ovat kuitenkin nykyään harvinaisia. (Smith, P. 2012.)

2.3 Apache

Apache on ilmainen, avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma. Apache on tehokas, joustava ja se noudattaa uusimpia protokollia mm. HTTP/1.1 (RFC2616). Apachessa on hyvät konfiguraatio- ja laajennus mahdollisuudet. Apache toimii useimmilla Linux, Windows ja Unix käyttöjärjestelmien versioilla. Apachea kehitetään jatkuvasti käyttäjien antamilla ideoilla ja palautteilla (The Apache Software Foundation 2013, hakupäivä 4.10.2013.)

Apache HTTP-palvelimen ensimmäinen virallinen versio 0.6.2 julkaistiin vuonna 1995. Versio 1.0 julkaistiin myöhemmin samana vuonna. Apachen uusin versio on 2.4.6, jonka vakaa versio on julkaistu 22.7.2013. Apachesta tuli vuodessa suosituin web-

palvelinohjelma, jota se on vielä tänäkin päivänä (The Apache Software Foundation 2013, hakupäivä 4.10.2013.)

2.3.1 Moduulit

Apachen suosio johtuu osittain siitä, että siinä on modulaarinen järjestelmä. Tämä mahdollistaa ulkopuolisten kehittäjien tehdä laajennuksia Apacheen. Moduuleita on monenlaisia esim. välityspalvelimeen ja tietoturvaan liittyviä.

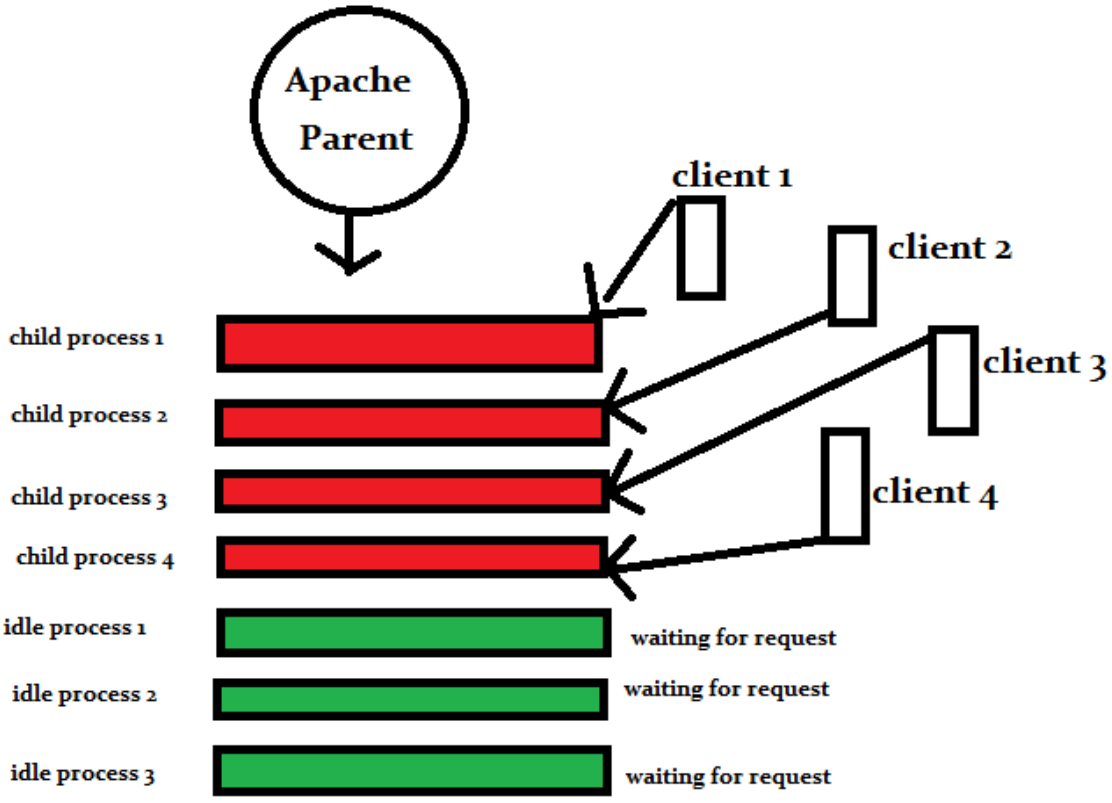
Jokaisessa moduulissa on kuitenkin huono puolikin, ne lisäävät muistin ja prosessorin käyttöä. Suorituskykyä haettaessa voidaan poistaa käyttämättömät moduulit käytöstä. Moduulien poistaminen ei ole kuitenkaan ihan helppoa, koska niitä on paljon. Ennen kuin moduuleita alkaa poistamaan käytöstä, kannattaa ottaa selville mitä ne tekevät. Debian-pohjaiset järjestelmät käyttävät omia tiedostoja jokaiselle moduulille, jotka löytyvät kansiosta /etc/apache2/mods-available (Smith, P. 2012.)

2.3.2 Moni-prosessi moduulit

Moni-prosessi moduulit (Multi-Process Modules, MPMs) kontrolloivat miten Apache käsittelee asiakkaiden pyyntöjä. Vain yksi MPM voi olla käytössä. Apachessa on kaksi moni-prosessi moduulia; Prefork ja Worker. Prefork on yleensä vakiona ja se on vakain. Workeria ei ole vielä testattu niin paljon, mutta se lupaa parempaa suorituskykyä (Smith, P. 2012.)

2.3.3 Prefork MPM

Prefork MPM:ssa on yksi hallintaprosessi, joka on vastuussa käynnistämään lapsiprosesseja, jotka hoitavat yhteyksiä. Lapsiprosessi hoitaa yhden yhteyden kerrallaan. Apache pyrkii pitämään aina useita vapaita prosesseja, jotka ovat valmiudessa palvelemaan tulevia pyyntöjä. Tällä tavoin asiakkaiden ei tarvitse odottaa uusien lapsiprosessien haarautumista (The Apache Software Foundation 2013, hakupäivä 4.10.2013)



KUVIO 1. Apache prefork MPM

Tässä on esimerkki preforkin asetuksista, jotka löytyvät /etc/apache2/apache2.conf tiedostosta:

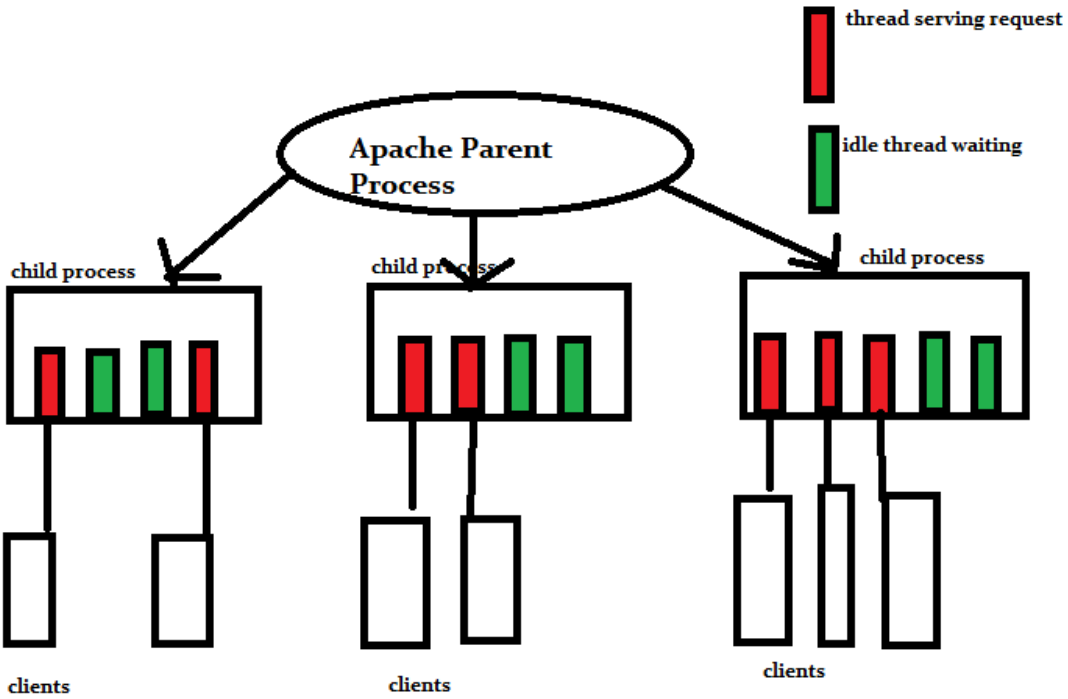
StartServers	10
MinSpareServers	5
MaxSpareServers	10
ServerLimit	100
MaxClients	100
MaxRequestsPerChild	1000

StartServers määrittelee, montako lapsiprosessia käynnistetään alussa. MinSpareServers ja MaxSpareServers kontrolloivat pienintä ja suurinta määrää vapaita lapsiprosesseja.

MaxClients määrittelee, kuinka monta lapsiprosessia voi olla päällä kerrallaan. ServerLimit tarkoittaa myös samaa, mutta sen asetusta ei voida vaihtaa pelkällä ”reload” komennolla. ServerLimit asetuksen vaihtaminen vaatii Apachen pysäyttämisen ja uudelleen käynnistyksen. MaxClientsin ja ServerLimitin ero on se, että ServerLimit auttaa Apachea arvioimaan paljonko keskusmuistia varataan Apachen käynnistymiseen. ServerLimitin tulee siis olla sama kuin MaxClients asetuksen tai vähän suurempi, jolloin MaxClients asetusta voidaan suurentaa ilman palvelimen uudelleen käynnistämistä (Smith, P. 2012.)

2.3.4 Worker MPM

Worker on muuten samanlainen kuin prefork, mutta se käyttää säikeitä. Workerin lapsiprosessit käyttävät säikeitä, joita voi olla useita yhdessä lapsiprosessissa. Yksi säie hoitaa yhden yhteyden. Konfiguraatio workerille on muuten saman tyylinen kuin prefork, mutta siinä määritellään, montako säiettä yksi lapsiprosessi voi ylläpitää. Worker on suunniteltu moniydin prosessoreille ja se skaalautuu paremmin kuin prefork (Smith, P. 2012.)



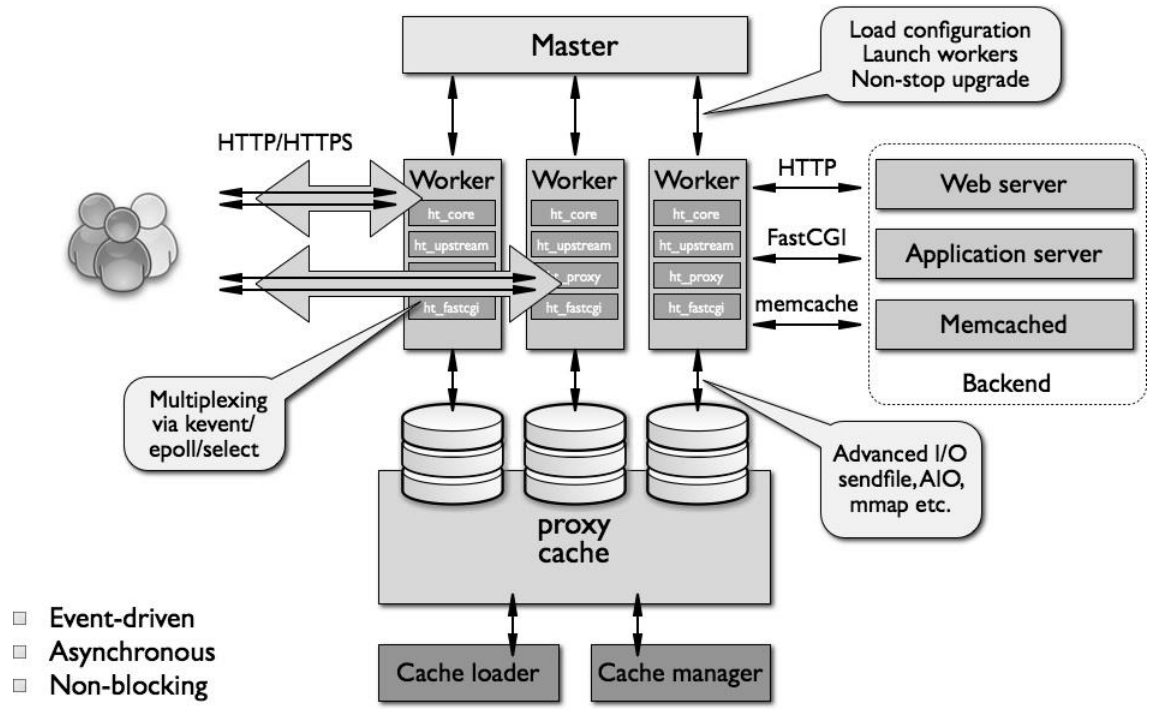
KUVIO 2. Apache worker MPM

2.4 Nginx

Nginx, joka lausutaan engine-x on ilmainen, avoimeen lähdekoodiin perustuva, suorituskykyinen HTTP-palvelin, välityspalvelin sekä sähköpostipalvelinohjelma. Igor Sysoev aloitti Nginxin kehittämisen 2002 ja julkaisi sen 2004. Nginx on käytössä 16.05% kaikista www-sivuista. Nginx tunnetaan sen suorituskyvystä, vakaudesta, monipuolisista ominaisuuksista, helposta konfiguraatiosta ja vähäisestä resurssien käyttämisestä (Nginx.org 2013, hakupäivä 4.10.2013.)

2.4.1 Event-driven

Yleensä web-palvelimissa käytetään prosesseihin tai säkeisiin (thread) perustuvaa tekniikaa. Nginx käyttää paremmin skaalautuvaa ”event-driven” arkkitehtuuria. Tämä arkkitehtuuri käyttää pieniä määriä muistia. Nginxin vahvuus on siinä, että se sopii käytettäväksi niin pieniin kuin suuriinkin palvelimiin (Nginx.org 2013, hakupäivä 4.10.2013.)



KUVIO 3. Nginx mestari ja työläiset

Nginxissä on yksi mestari prosessi, joka delegoi tehtäviä yhdelle tai useammalle työläiselle. Työläisten määrä voidaan määritellä Nginxin asetustiedostossa. Yksi työläinen voi käsitellä tuhansia pyyntöjä (Alexeev, A. 2013, hakupäivä 14.11.2013.)

2.4.2 Asetustiedosto

Tässä Nginxin asetustiedoston alkuosa, jossa on tärkeimmät asetukset:

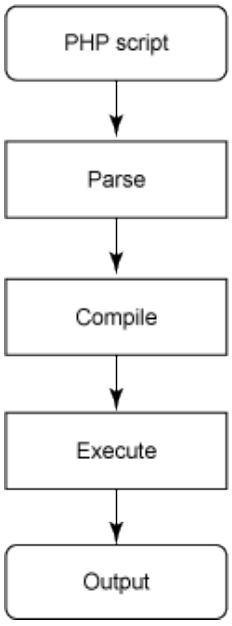
```
user www-data;
worker_processes  1;
error_log  /var/log/nginx/error.log;
pid  /var/run/nginx.pid;
events {
  worker_connections  768;
}
http {
  include  /etc/nginx/mime.types;
  default_type  application/octet-stream;
  access_log  /var/log/nginx/access.log;
  sendfile  on;
  keepalive_timeout  65;
  tcp_nodelay  on;
  gzip  on;
  include /etc/nginx/sites-enabled/*;
}
```

Worker_processes määrittelee työläisten määrän. Työläisten määräksi kannattaa asettaa sama kuin tietokoneen prosessorien määrä. Worker_connections määrittelee montako samanaikaista käyttäjää yksi työläinen voi palvella. Samanaikaisten käyttäjien määrä on siis worker_processes * worker_connections. Error_log ja access_log ilmoittavat logien sijainnin. Keepalive_timeout määrittelee kauanko ei toiminnallinen yhteys kestää asiakkaan ja palvelimen välillä, ennen kuin se suljetaan. Sendfile sallii Nginxin käyttää erityistä järjestelmäkutsua lähettääkseen tiedostoja verkkoon tehokkaasti. Gzip pakkaa lähetettävät tiedostot (Alexeev, A 2013, hakupäivä 14.11.2013.)

Nginx on käytössä suurissa sivustoissa muun muassa: Netflix, Hulu, Pinterest, CloudFlare, Airbnb, WordPress.com, GitHub, SoundCloud, Zynga, Eventbrite, Zappos, Media Temple, Heroku, RightScale, Engine Yard and NetDNA (Nginx.org 2013, hakupäivä 13.11.2013.)

2.5 PHP

PHP on HTML:ään sulautettu skriptikieli. Suurin osa sen syntaksista on lainattu C-kielestä, Javasta ja Perlistä. Kielen tarkoituksena on auttaa web-kehittäjiä kirjoittamaan dynaamisia sivuja nopeasti. Lyhenne PHP tulee sanoista Hypertext Preprocessor. Lyhenne ei täsmää, koska se on rekursiivinen akronyymi, joka on hakkerien suosima humoristinen tapa viitata heihin itseensä, toisiin akronyymeihin tai lyhenteisiin (PHP.net. 2013, hakupäivä 20.10.2013.)

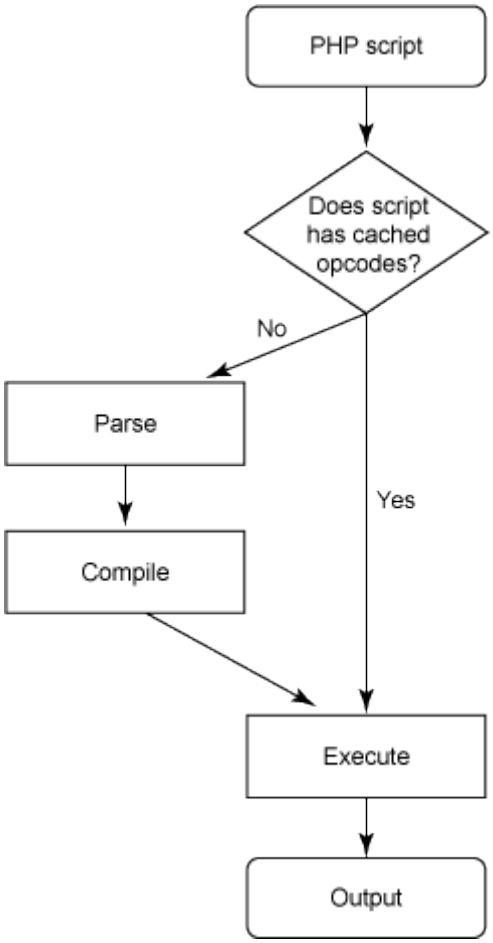


KUVIO 4. PHP-koodin toiminta ((Mertic, J 2011, hakupäivä 13.11.2013.)

2.5.1 Opcode välimuisti

Operation code (opcode) on koodia, joka antaa tietokoneelle toiminnon, jonka se suorittaa. Opcode sisältää numeroita. Helpoin tapa parantaa PHP:ta sisältävien web-sivujen suorituskykyä HTTP-palvelimessa on käyttää opcode välimuistia. Yleensä suorituskyky paranee vähintään puolella.

PHP on tulkattu ohjelmointi kieli, toisin kuin C tai Java, jotka ovat käännettyjä. Tämän takia jokaisessa pyynnössä käsitellään jäsentely, käännökset ja ajo. Tämä on resursseja ja aikaa vievä tapa, ottaen huomioon, että skripti harvoin vaihtuu pyyntöjen välissä. Kun skripti on jäsennelty ja käännetty, skripti koostuu sarjoista opcodeja. Tässä vaiheessa opcode välimuisti tulee mukaan. Se tallentaa välimuistiin toimintokoodit välttääkseen jäsentelyn ja kääntämisen jokaiselta pyynnöllä (Mertic, J 2011, hakupäivä 13.11.2013.)



KUVIO 5. Opcode välimuistin hyöty PHP:ssa

Niin kuin kuvassa näkyy, voidaan opcodet ajaa suoraan. Tarkistusalgorytmin avulla voidaan hoitaa tilanteet, joissa skiptiin on tullut muutoksia. Kun muutoksia tulee, opcodet päivitetään (Mertic, J 2011, hakupäivä 13.11.2013.)

2.5.2 Wordpress

Wordpress on 2003 aloitettu ohjelmointi-projekti, jolla oli vain pieni käyttäjämäärä. Tarkoituksena oli parantaa typografiaa joka päiväisessä kirjoittamisessa. Ajan kuluessa se on kasvanut suurimmaksi bloggaus-välineeksi maailmassa, jota käytetään miljoonilla sivuilla ja joita selaa 10 miljoonaa ihmistä päivittäin (Wordpress 2013a, hakupäivä 25.10.2013.)

Wordpress on täysin ilmainen avoimen lähdekoodin ohjelma. Wordpress on alunperin tehty bloggaukseen, mutta nykyään sillä on mahdollista tehdä erilaisia sivuja monien liitännäisten avulla. Ohjelma asennetaan omalle verkkopalvelimelle. Sen voi ladata Wordpress.org kotisivuilta. Wordpressin vaatimuksia verkkopalvelimelta ovat PHP 5.2.4 tai uudempi, MySQL 5.0 tai uudempi ja HTTP-palvelinohjelmaksi suositellaan Apachea tai Nginxiä (Wordpress 2013a, hakupäivä, 25.10.2013), (Wordpress 2013b, hakupäivä, 25.10.2013.)

2.5.3 WP Super Cache

WP Super Cache on Wordpressin liitännäinen, joka muuntaa dynaamiset Wordpress tiedostot staattiseksi. Kun html-tiedosto on luotu, HTTP-palvelin alkaa käyttämään tätä tiedostoa sen sijaan, että ajaisi hitaampia Wordpressin PHP-skriptejä. Staattisia sivuja käyttävät anonyymit käyttäjät, eli sellaiset jotka eivät ole kirjautuneet (Wordpress 2013c, hakupäivä 13.11.2013.)

3 RASPBERRY PI

Raspberry Pi on luottokortin kokoinen tietokone, joka toimitetaan yleensä ilman koteloa. Raspberry Pi:stä on A- ja B-malli. A-mallissa on yksi USB-liitäntä, 256 MB keskusmuistia ja siinä ei ole verkkoliitäntää. B-mallissa on kaksi USB paikkaa, sekä 512 MB keskusmuistia. Raspberry Pi jakaa keskusmuistin prosessorin ja näytönohjaimen kanssa. Näytönhohjain on tarpeeksi tehokas pyörittämään Blu-ray tason kuvaa (McManus, S & Cook, M 2013, hakupäivä 6.10.2013.)

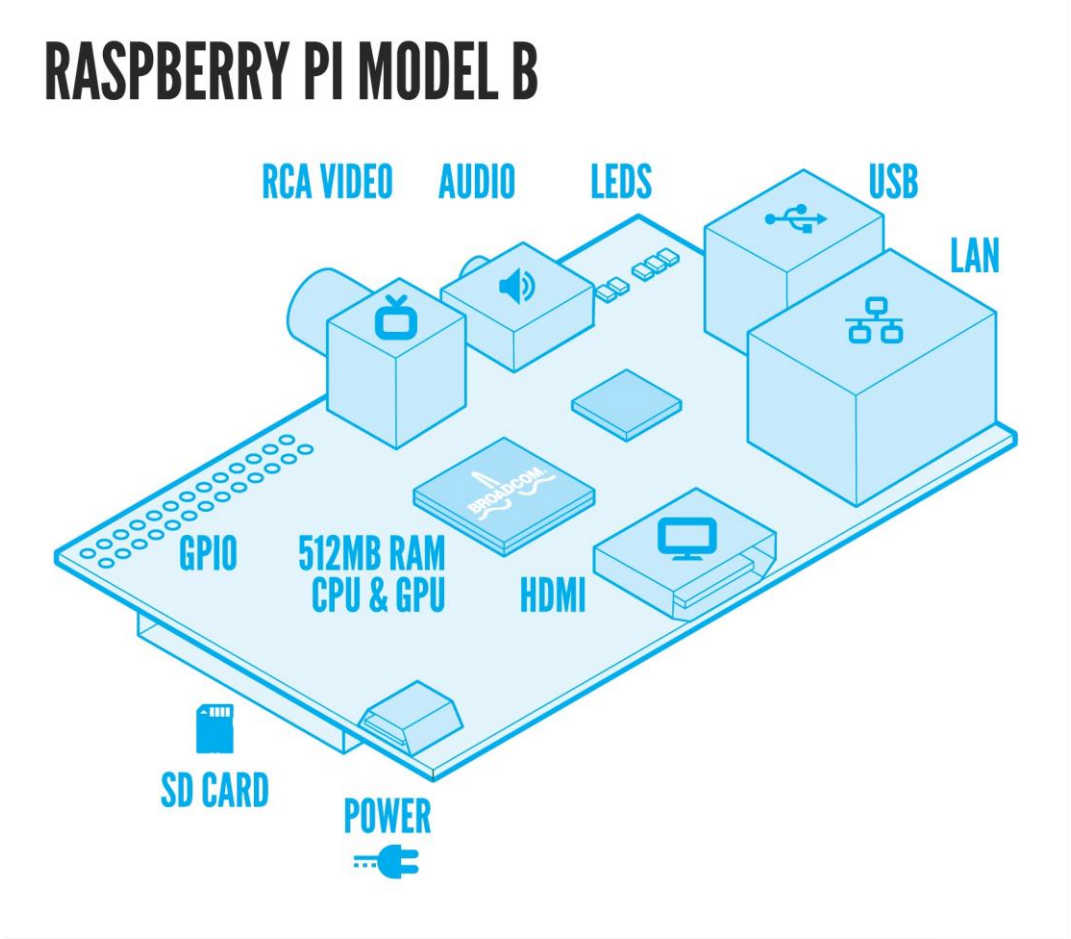
Raspberry Pi:lle optimoitu Raspbian käyttöjärjestelmä käynnistyy komentoriville, mutta siinä on myös mahdollista käyttää graafistatyöpöytää. Raspberry Pi:llä voi tehdä kaikki normaalit tietokoneella tehtävät asiat muun muassa Internetin selaaminen, tekstinkäsittely, taulukkolaskenta ja kuvankäsittely. Myös musiikin kuuntelu, videon katsominen ja pelien pelaaminen onnistuu laitteella. Raspberry Pi on edullinen vaihtoehto HTPC:ksi, eli kotiteatteri-tietokoneeksi. Siinä toimii XBMC mediantoisto-ohjelma, jonka avulla voi katsoa videoita, jotka sijaitsevat kotiverkossa. Raspberry Pi:tä voidaan käyttää lukuisissa muissa käyttötarkoituksissa esim. erilaisina palvelimina, nettiradiona tai vaikka nettikameran tietokoneena. Raspberry Pi on kuitenkin parhaimmillaan oppimiskäytössä miten tietokoneet toimivat ja miten niitä ohjelmoidaan (McManus, S & Cook, M 2013, hakupäivä 6.10.2013.)

General Purpose I/O (GPIO) ansiosta Raspberry Pi:tä voidaan käyttää myös elektroniikka projekteissa. GPIO on yleiskäyttöinen portti mikrokontrollereissa ja mikroprosessoreissa. GPIO voidaan ohjelmoida joko signaalin vastaanottajaksi tai lähettäjäksi (McManus, S & Cook, M 2013, hakupäivä 6.10.2013.)

3.1 Tekniset ominaisuudet

Raspberry Pi B-malli:

Prosessori	Broadcom BCM2835 700 MHz ARM11 ARM1176JZF-S core
Näytönohjain	Broadcom VideoCore IV,OpenGL ES 2.0,OpenVG 1080p30 H.264
Keskusmuisti	512 MB (SDRAM)
USB portit	2
Video ulostulo	Komposiitti video | Komposiitti RCA, HDMI
Ääni ulostulo	3.5 mm jakki, HDMI
Muisti	SD-kortti tai ulkoinen kovalevy
Verkkoliitäntä	10/100 Ethernet RJ45
Virtalähde	5 V Mircro USB 700 mA, (3.5 W)
Koko	85.0 x 56.0 mm x 17mm
Paino	40g



(Elinux 2012, hakupäivä 6.10.2013),(Raspberry 2013, hakupäivä 16.12.)

3.2 Raspbian

Raspberry Pi:lle suositellaan käyttöjärjestelmäksi Raspbiania, joka on Debianiin perustuva käyttöjärjestelmä. Se on kevyt käyttöjärjestelmä, joka suunniteltu Raspberry Pi:lle käytettäväksi. (McManus, S & Cook, M. 2013, hakupäivä 6.10.2013)

Käyttöjärjestelmässä on perustyökalut ja ohjelmat. Raspbianin mukana tulee yli 35000 ohjelmistopakettia, jotka on optimoitu sopivaksi Raspberry Pi:n suorituskyvylle. Ohjelmistopaketit lisättiin kesällä 2012. Raspbianin ohjelmistopaketteja kehitetään kuitenkin vielä erityisesti vakauden ja suorituskyvyn kannalta. (Raspbian, hakupäivä 7.10.2013)

Raspbian käyttöjärjestelmää ei kehittänyt The Raspberry Pi Foundation, vaan pieni omistautunut tiimi kehittäjiä, jotka olivat kiinnostuneet Raspberry Pi:stä ja sen mahdollisuuksista koulutuksessa. The Raspberry Pi Foundation on kuitenkin julkaissut heidän oman versionsa Raspbianista. (Raspbian, hakupäivä 7.10.2013)

4 ASENNUKSET

Tässä kappaleessa kerrotaan lyhyesti ohjelmistojen asennuksista. Asennukset tehdään toisella tietokoneella käyttämällä putty.exe ohjelmaa, jolla luodaan etäyhteys palvelimeen. Käytettävät tietokoneet ovat samassa lähiverkossa.

4.1 Raspberry Pi

Ladataan wheezy-raspbian käyttöjärjestelmän image-tiedosto Raspberry Pi:n kotisivuilta http://www.raspberrypi.org/downloads. Windowsin käyttäjille suositellaan asennettavaksi Win32DiskImager ohjelma, jolla Raspbianin image-tiedosto kirjoitetaan SD-kortille, jonka koko on 2 GB tai suurempi. UNIX käyttäjät voivat kirjoittaa sen dd-ohjelmalla, jonka latauslinkki löytyy myös Raspberryn sivuilta.

Alustetaan uusi SD-kortti. Käynnistetään Win32DiskImager, valitaan Raspbianin image-tiedosto ja kirjoitetaan se SD-kortille. Tämän jälkeen SD-kortti voidaan kytkeä Raspberry Pi:n SD-korttipaikkaan. Tämän jälkeen Raspberry Pi:n virtajohto voidaan kytkeä ja Raspbian käyttöjärjestelmä käynnistyy komentoriville.

4.2 LAMP ja Wordpress

LAMP lyhenne tulee sanoista: Linux, Apache, MySQL ja PHP. Asennetaan Apache komennolla:

```
apt-get install apache2
```

Ohjelma asentaa Apachen 2.2 version.

Kun asennus on valmis, voidaan toimivuus testata toisella tietokoneella kirjoittamalla verkkoselaimen osoiteriville Raspberry Pi:n IP-osoite, jonka saa selville kirjoittamalla komentoriville ”ifconfig”. Apachen aloitussivu käynnistyy, mikäli Apachen asennus on onnistunut ja lähiverkkoyhteydet ovat kunnossa.

Asennetaan PHP komennolla:

```
sudo apt-get install php5 php5-mysql
```

Luodaan /var/www kansioon info.php tiedosto, jonka sisällöksi tallennetaan teksti:

```
<?php phpinfo (); ?>
```

Käynnistetään Apache:

```
sudo service apache2 restart
```

Testataan toimivuus verkkoselaimella osoitteessa IP-osoite/info.php

Asennetaan MySQL komennolla:

```
sudo apt-get install mysql-server
```

Asennuksessa luodaan pääkäyttäjän salasana tietokannalle.

Asennuksen jälkeen aktivoidaan tietokanta komennolla:

```
sudo mysql_install_db
```

Ajetaan skripti, jolla päästään määrittelemään asetuksia:

```
sudo /usr/bin/mysql_secure_installation
```

Poistetaanko anonyymit käyttäjät? Kyllä

Kielletäänkö pääkäyttäjän kirjautuminen etänä? Ei

Poistetaanko testi tietokanta? Kyllä

Otetaanko tehdyt muutokset heti käyttöön? Kyllä

Luodaan uusi MySQL tietokanta ja käyttäjä, jolle annetaan salasana. MySQL komennot päättyvät puolipisteesen.

Kirjaudutaan MySQL:

```
mysql –u root –p
```

Luodaan tietokanta wordpress:

```
CREATE DATABASE wordpress;
```

Luodaan käyttäjä:

```
CREATE USER tero@localhost;
```

Salasana käyttäjälle:

```
SET PASSWORD FOR tero@localhost= PASSWORD(“password”);
```

Annetaan kaikki oikeudet käyttäjälle:

```
GRANT ALL PRIVILEGES ON wordpress.* TO tero@localhost IDENTIFIED
BY ’password’;
```

Päivitetään oikeudet:

```
FLUSH PRIVILEGES;
```

Poistutaan MySQL:

```
exit
```

Uudelleen käynnistetään Apache2:

```
sudo service apache2 restart
```

Ladataan Wordpress:

```
sudo wget http://wordpress.org/latest.tar.gz
```

Puretaan paketti, joka ladattiin käyttäjän kotikansioon:

```
tar –xzvf latest.tar.gz
```

Wordpress kansiossa tehdään kopio sample tiedostosta, jolle tehdään konfiguraatiot:

```
sudo cp wp-config-sample.php wp-config.php
```

Muokataan wp-config.php tiedostoa ja määritellään sinne MySQL:n Wordpress tietokanta, käyttäjä ja salasana. Siirretään wordpress kansio kotisivukansioon:

```
sudo mv wordpress /var/www/
```

Suoritetaan asennus loppuun verkkoselaimella osoitteessa: palvelimen ip-osoite/wp-admin/install.php

4.3 LEMP ja Wordpress

LEMP lyhenne tulee sanoista: Linux, Engine x (Nginx), MySQL ja PHP. MySQL:n asennus tapahtuu samalla tavalla, kuin Apachessa. Tässä työssä asennettu Nginx versio on 1.2.1.

Asennetaan nginx komennolla:

```
sudo apt-get install nginx
```

Käynnistetään nginx komennolla:

```
sudo service nginx start
```

Asennetaan php5-fpm ja php5-mysql:

```
sudo apt-get install php5-fpm php5-mysql
```

Muokataan php.ini tiedostoa:

```
sudo nano /etc/php5/fpm/php.ini
```

Käynnistetään php5-fpm:

```
sudo service php5-fpm restart
```

Luodaan info.php tiedosto nginxin www kansioon:

```
sudo nano /usr/share/nginx/www/info.php
```

info.php tiedostoon tallennetaan sama teksti kuin Apache asennuksessa:

```
<?php

phpinfo();
```

```
?>
```

Käynnistetään nginx:

```
sudo service nginx restart
```

Tehdään Wordpress virtual host tekemällä uusi kopio tiedostosta:

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/wordpress
```

Muokataan tiedostoa:

```
sudo nano /etc/nginx/sites-available/wordpress
```

Muutetaan tiedoston alkuosa:

```
server {
        listen   80;


        root /var/www/wordpress;
        index index.php index.html index.htm;
        server_name 192.168.0.102;
        location / {
                try_files $uri $uri/ /index.php?q=$uri&$args;
        }
        error_page 404 /404.html;
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
              root /usr/share/nginx/www;
        }
        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
        location ~ \.php$ {
                try_files $uri =404;
                #fastcgi_pass 127.0.0.1:9000;
```

```
                # With php5-fpm:
                fastcgi_pass unix:/var/run/php5-fpm.sock;
                fastcgi_index index.php;
                include fastcgi_params;
                 }

}
```

Vaihdetaan server_name vastaamaan palvelimen osoitetta. Juurikansioksi määritellään var/www.

Poistetaan alkuperäinen default tiedosto:

```
sudo rm /etc/nginx/sites-enabled/default
```

Tehdään symbolinen Wordpress linkki sites-available -> sites-enabled:

```
sudo ln -s /etc/nginx/sites-available/wordpress /etc/nginx/sites-enabled/wordpress
```

5 OPTIMOINTI

Tässä kappaleessa tehdään suorituskykyä parantavia optimointeja Raspberry Pi:lle ja HTTP-palvelinohjelmille. Optimoinnit tehdään toisella tietokoneella, käyttämällä putty.exe ohjelmaa, jolla luodaan etäyhteys palvelimeen.

5.1 Raspberry

Kun Raspberry Pi käynnistyy ensimmäistä kertaa Raspbianiin, järjestelmällä on käytössä 59 MB keskumuistia. Seuraavien asennuksien jälkeen keskumuistin käyttö järjestelmän käynnistyessä on 41 MB. Seuraavat komennot on ajettu pääkäyttäjänä komentorivillä.

Ajetaan komento ”raspi-config”, josta valitaan ”Expand Filesystem”. Näin saadaan kaikki muisti käyttöön SD-kortilta, joka olisi muuten vain 2 GB kortin koosta riippumatta. Raspi-config ikkunassa valitaan myös ”Advanced Options” ja vaihdetaan ”Memory Split” kohdassa 16 MB muistia näytönohjaimen käyttöön, koska graafista työpöytää ei tarvitse käyttää palvelimella. Keskusmuistia on tämän jälkeen 496 MB.

Päivitetään Raspbian ja poistetaan asennuksiin liittyvät ylimääräiset tiedostot komennolla:

```
apt-get -y update && apt-get -y dist-upgrade && apt-get -y autoremove &&
apt-get -y autoclean
```

Poistetaan työpöytä ohjelmat, työkalut ja ikkunanhallinnoijan:

```
apt-get purge consolekit desktop-base* desktop-file-utils* gnome-icon-
theme* gnome-themes-standard* hicolor-icon-theme* leafpad* lxde* lxde-
core* midori* xserver-common* xserver-xorg* xserver-xorg-core* xserver-
xorg-input-all* xserver-xorg-input-evdev* xserver-xorg-input-synaptics*
xserver-xorg-video-fbdev*
```

Useimmissa Linux käyttöjärjestelmissä on käytössä OpenSSH ohjelma etäyhteyksiä varten. On kuitenkin olemassa kevyempiä etäyhteyden luomiseen käytettäviä ohjelmia

mm. Dropbear. Seuraavaksi asennetaan Dropbear siten, että säilytämme etäyhteyden palvelimeen.

Asennetaan Dropbear:

```
apt-get install dropbear
```

Muokataan Dropbearin tiedostoa niin, että se käynnistyy järjestelmän käynnistyessä:

```
sed -i 's/NO_START=1/NO_START=0/g' /etc/default/dropbear
```

Pysäytetään nykyinen SSH palvelin, jolla vapautamme 22 portin Dropbearin käyttöön. Tämän hetkinen etäyhteys ei katkea:

```
/etc/init.d/ssh stop
```

Käynnistetään Dropbear:

```
/etc/init.d/dropbear start
```

Poistetaan OpenSSH:

```
apt-get remove openssh-server
```

Vaihdetaan Bash komentorivi Dashiin, joka on kevyempi:

```
dpkg-reconfigure dash
```

Poistetaan ylimääräiset tty:t ja gettyn, joita tarvitaan kun luodaan uusia terminaali yhteyksiä:

```
sed -i '/[2-6]:23:respawn:\/sbin\/getty 38400 tty[2-6]/s%^%#%g' /etc/inittab
```

5.2 Apache

Asennetaan Alternative PHP Cache:

```
sudo apt-get install php-apc
```

Uudelleen käynnistetään Apache:

```
sudo service apache2 restart
```

Wordpress liitännäisten asentamiseen sivusto kysyy ftp-palvelimen tietoja liitännäisen lataamista varten. Tämä johtuu siitä, että apachen asetustiedostoon ei ole määritelty käyttäjää. Määritellään käyttäjä ja ryhmä, jolla Apache ja Wordpress on asennettu:

Tämän jälkeen Wordpressillä on oikeus kirjoittaa kansioihinsa ja ladata liitännäisiä.

Apachen konfiguraatio tiedostoon tehtiin prefork kohtaan pieniä muutoksia seuraaviin asetuksiin:

StartServers	3
MinSpareServers	3
MaxSpareServers	10
MaxClients	50
KeepAliveTimeout	5

“ps -ylC apache2 --sort:rss” komennolla selvitettiin Apachen prosessien koko, joka oli keskimäärin 7 MB. Sopivaksi Maxclients määräksi saatiin 7 * 50 = 350. Tämä ei ole tarkka arvio, mutta kertoo suunnilleen paljon Maxclients arvoksi voidaan laittaa Raspberry Pi:n muistin määrään nähden.

5.3 Nginx

5.3.1 nginx.conf

nginx.conf tiedoston perus asetukset:

```
worker_processes 1;
pid /var/run/nginx.pid;
events {
        worker_connections 768;
        # multi_accept on;
}
```

```
http {
        ##
        # Basic Settings
        ##
        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 5;
        types_hash_max_size 2048;
        # server_tokens off;
        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;
        include /etc/nginx/mime.types;
        default_type application/octet-stream;
        ##
        # Logging Settings
        ##
        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;
        ##
        # Gzip Settings
        ##
        gzip on;
        gzip_disable "msie6";
```

Asetetaan worker_processes määräksi 1, koska Raspberry Pi:ssä on vain 1 prosessori. Prosessorien määrän voi tarkastaa komennolla:

```
pi@raspberrypi ~ $ grep processor /proc/cpuinfo | wc -l

1
```

Keepalive_timeout on muutetttu 65 -> 5. Tällä asetuksella yhteys suljetaan 5 sekunnin kuluttua jos se ei ole aktiivinen.

Asennetaan Alternative PHP Cache komennolla:

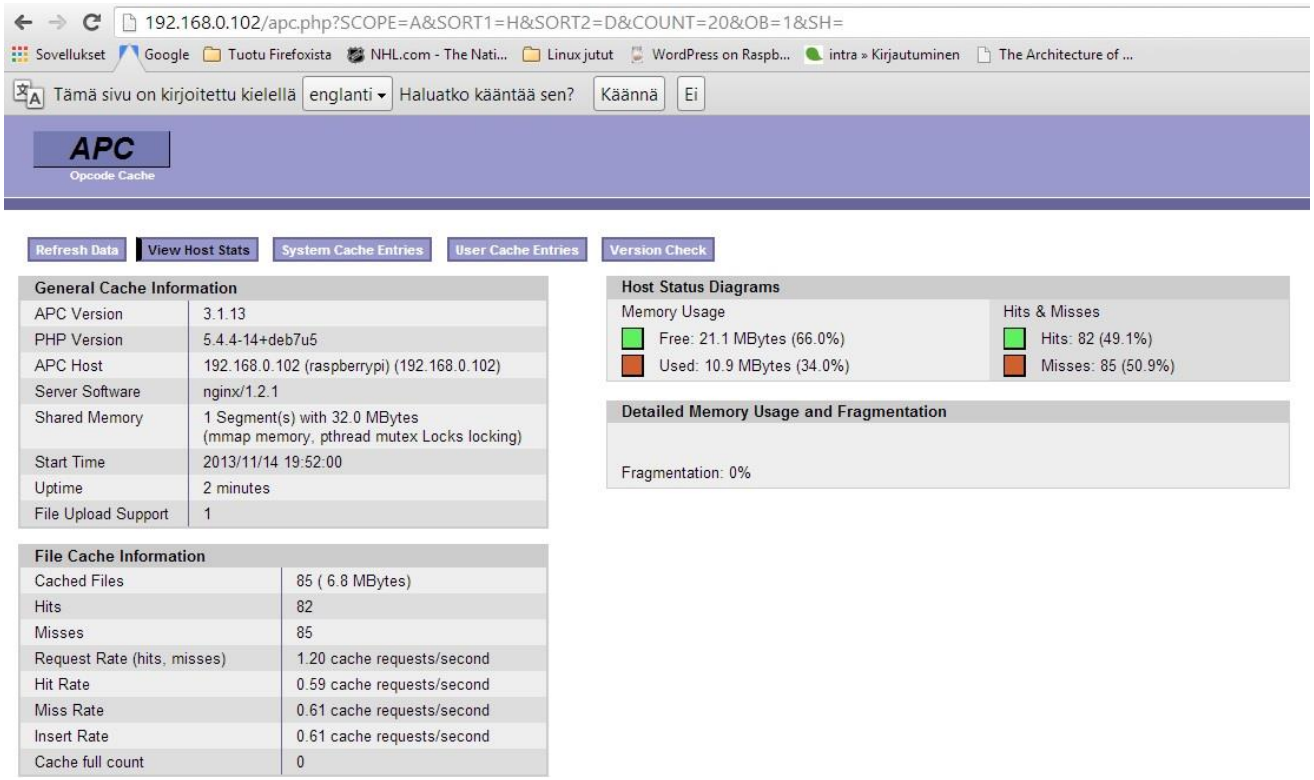
```
sudo apt-get install php-apc
```

APC alkaa toimia heti nginxin uudelleen käynnistyksen jälkeen. APC:n keskusmuistin käyttöä voidaan säätää sen konfiguraatio tiedostossa kohdassa apc.shm_size, jonka arvoksi on vakiona annettu 32 MB. Tässä optimoinnissa ei ole tarvetta muuttaa asetusta, koska testattavien www-sivujen koko ei ole niin suuri.

Kopioidaan tiedosto apc.php kotisivu kansioon:

```
pi@raspberrypi ~ $ sudo cp /usr/share/doc/php-apc/apc-php /var/www
```

Nyt voidaan tarkastella APC:n muistin käyttöä verkkoselaimella osoitteessa palvelimenosoite/apc.php. Tässä kuvankaappauksessa nähdään tallennettujen tiedostojen määrä välimuistissa ja niiden koko megabitteinä.

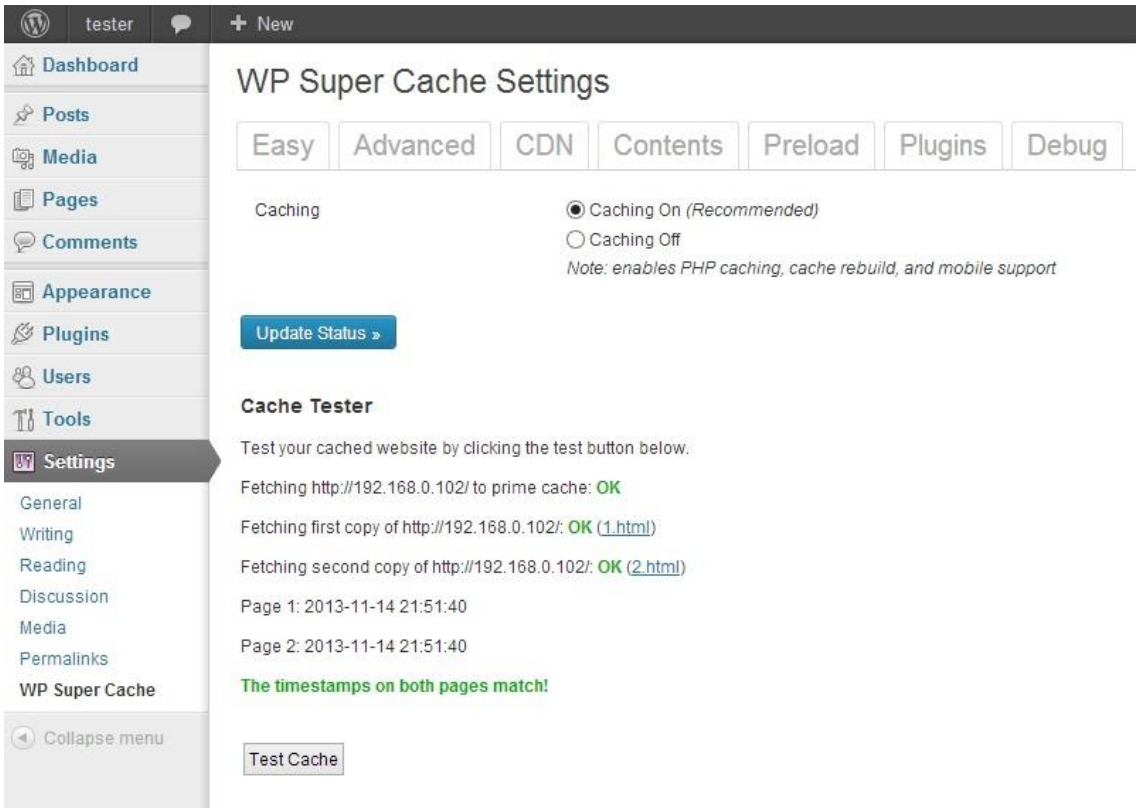


Välimuistiin tallennettujen sivujen määrä on 6,8 megabittia. Ei ole siis tarvetta nostaa muistin määrää 32 MB:stä suuremmaksi.

WP Super Cache asennetaan kirjautumalla Wordpressin sivuille. WP Super Cache vaatii permalink asetuksen olevan ”Post name”. Asetuksen voi muuttaa kohdassa Settings -> Permalinks. WP Super Cache löytyy käyttämällä hakua kohdassa Plugins -> Add New.

Klikataan ”Install now”, jolloin ohjelma ladataan ja asennetaan. Lopuksi vaihdetaan ”Caching On” päälle ohjelman asetuksista.

WP Super Cachen toimivuuden voi testata heti samalla sivulla:



Toimivuustesti ajettiin ja ohjelma ilmoittaa toimivansa.

6 TESTAUS JA JOHTOPÄÄTÖKSET

6.1 Siege

Siege on avoimeen lähdekoodiin perustuva rasitustestiohjelma ja suorituskyvyn mittaaja. Siege on kehitetty web-kehittäjille ja ylläpitäjille, jonka avulla he voivat testata sivujaan ja järjestelmiään. Se voi testata yksittäistä URL:ia käyttäjän määrittämällä käyttäjämäärällä tai lukea monta URL:ia muistiin ja testata niitä samanaikaisesti. Ohjelma ilmoittaa kaikkien tapahtumien määrän, siirretyt bitit, vastausaika, käytetty aika, samanaikaiset käyttäjät ja onnistuneiden vastauksien määrä. Useimmat testin ominaisuudet on mahdollista asettaa haluamakseen komentorivillä (Fulmer, J 2012, hakupäivä 7.10.2013).

Siegessä testejä ajetaan kirjoittamalla komentoriville Siege ja käyttämällä lyhenteitä, joista yleisimmät ovat:

-c = Samanaikaiset käyttäjät

Esimerkki: ”–c10” = 10 käyttäjää”

-t = Testin pituus ja aikamääre sekunteina S, minuutteina M, tunteina H.

Esimerkki: ”–t10S” = 10 sekunnin mittainen testi

-d = Jokaiselle siege käyttäjälle laitetaan viivettä satunnainen määrä väliltä 1 ja –d asetettu määrä. Oletuksena –d on 3, joten viive arvotaan välille 1-3 sekuntia. Kun haluamme saada tarkempia suorituskyky tuloksia, asetamme arvoksi 1 (-d1). Tällä tavoin viive on aina 1 sekuntia.

Esimerkki komentoriville kirjoitettavasta lauseesta:

```
sudo siege –c100 –t15S –d1 192.168.1.35
```

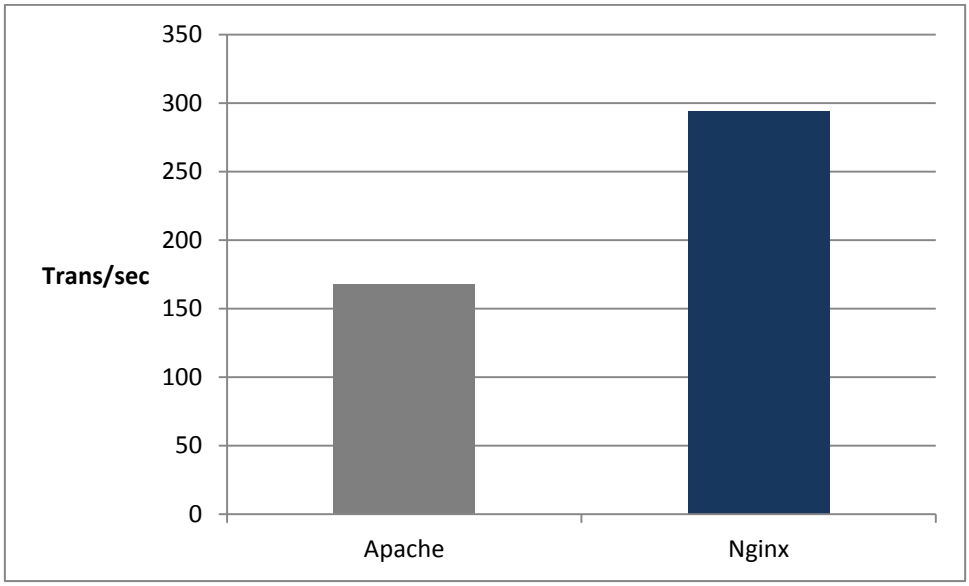
6.2 Staattiset sivut

Testataan staattisia HTML-sivuja, joissa on kolme .jpg kuvatiedostoa ja 10 riviä tekstiä. Sivujen koko on 168 kilotavua, jonka näkee komennolla ”du –hs /var/www”. Web-palvelimet on asennettu oletusasetuksilla. Ajetaan siege testi 200 käyttäjää samanaikaisesti 1 sekunnin viiveellä. Testi kestää yhden minuutin.

```
sudo siege -c200 -d1 -t1M 192.168.0.103
```

siege -c200 -d1 -t1M	Apache	Nginx
Transactions:	10066 hits	17545 hits
Availability:	100.00 %	100.00 %
Elapsed time:	59.75 secs	59.65 secs
Data transferred:	10.65 MB	19.24 MB
Response time:	0.67 secs	0.18 secs
Transaction rate:	168.47 trans/sec	294.13 trans/sec
Throughput:	0.18 MB/sec	0.32 MB/sec
Concurrency:	112.29	52.35
Successful transactions:	10066	17545
Failed transactions:	0	0
Longest transaction:	2.82	0.38
Shortest transaction:	0.00	0.00

TAULUKKO 1. Staattiset sivut



Tässä ja seuraavissa taulukoissa on kuvattu tapahtumien määrä, joka tarkoittaa palveltujen asiakkaiden määrää sekunnissa. Dokumentoituna on kaikki testin tulokset, jotka Siege tulostaa, mutta trans/sec arvolla on hyvä vertailla nopeuksia.

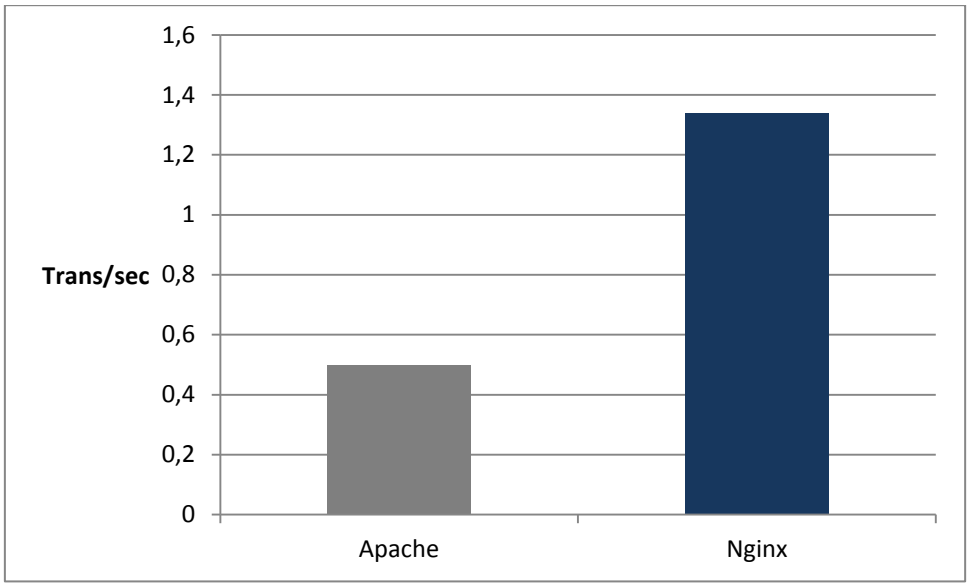
6.3 Wordpress-sivut

Ajetaan testi Wordpress-sivuille, jotka on asennettu perusasennuksella. Testataan kolmella samanaikaisella käyttäjällä, koska seuraavassa testissä käytettävä 50 käyttäjää aiheutti palvelimen kaatumisen.

```
sudo siege -c3 -d1 -t1M 192.168.0.102/wordpress
```

siege -c3 -d1 -t1M	Apache	Nginx
Transactions:	30 hits	80 hits
Availability:	100.00 %	100.00 %
Elapsed time:	59.48 secs	59.79 secs
Data transferred:	0.04 MB	0.12 MB
Response time:	4.80 secs	1.64 secs
Transaction rate:	0.50 trans/sec	1.34 trans/sec
Throughput:	0.00 MB/sec	0.00 MB/sec
Concurrency:	helmi.42	helmi.19
Successful transactions:	33	82
Failed transactions:	0	0
Longest transaction:	loka.55	kesä.13
Shortest transaction:	0.00	0.34

TAULUKKO 2. Wordpress-sivut



Nginxin nopeus ero oli suurempi kuin staattisilla sivuilla. Testissä selviää myös, ettei Raspberry Pi:n tehot riitä palvelemaan Wordpress sivuja ilman optimointia.

6.4 Optimoidut Wordpress-sivut

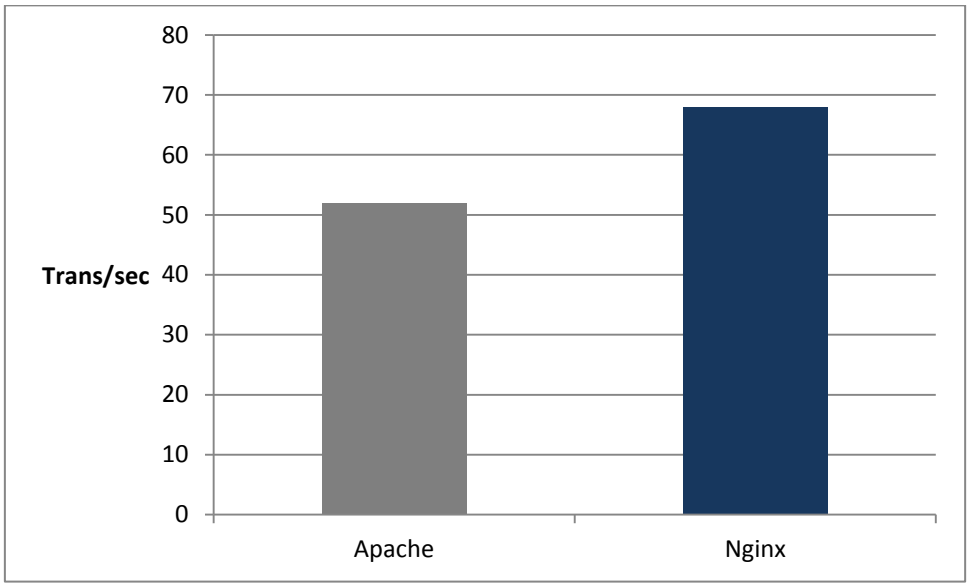
Tässä testissä palvelimille ajetaan minuutin mittainen testi 50:lla käyttäjällä. Palvelimille on tehty pieniä muutoksia konfiguraatio tiedostoon, jotka on kerrottu optimointi osiossa sekä asennettu opcode cache APC ja Wordpress liitännäinen Super WP Cache. WP Super Cachen asetukset ovat niin, että sivuja ei tallenneta täysin staattiseksi, vaan niihin jää jonkin verran PHP-koodia. Tämän takia APC:stä on myös hyötyä. Testissä ajetaan Wordpress-sivujen etusivua, joka on muuten alkuperäinen, mutta siihen on lisätty kalenteri.

Ajetaan testi siege komennolla:

```
sudo siege -c50 -d1 -t1M 192.168.0.102/wordpress
```

siege -c50 -d1 -t1M	Apache	Nginx
Transactions:	3120 hits	4016 hits
Availability:	100.00 %	100.00 %
Elapsed time:	59.99 secs	59.43 secs
Data transferred:	4.49 MB	5.32 MB
Response time:	0.46 secs	0.23 secs
Transaction rate:	52.01 trans/sec	67.58 trans/sec
Throughput:	0.07 MB/sec	0.09 MB/sec
Concurrency:	23.70	15.51
Successful transactions:	3152	4048
Failed transactions:	0	0
Longest transaction:	maalis.59	tammi.14
Shortest transaction:	0.00	0.00

TAULUKKO 3. Optimoidut Wordpress-sivut



Optimoitujen Wordpress-sivujen testissä nopeusero ei ole niin suuri kuin täysin staattisilla sivuilla tai perus Wordpress asennuksessa. Apache hyötyi enemmän optimoinnista, mutta Nginx oli silti edelleen nopeampi. Top-työkalulla kun tarkkaili prosessorin ja muistin käyttöä, huomasi että prosessorin tehot loppuivat ennen keskusmuistia.

7 POHDINTA

Opinnäytetyön aihe oli mielestäni hyvä, mutta myös haastava. Halusin tehdä opinnäytetyön liittyen palvelimiin. Opinnäytetyö on minusta myös ajankohtainen, koska Raspberry Pi on aika uusi tietokone ja niitä myydään aktiivisesti. Raspberry Pi sopii mainiosti HTTP-palvelimeksi kotikäyttöön, sen vähäisen hinnan ja virran kulutuksen takia.

Työn alkuvaiheessa materiaalia täytyi opiskella aika paljon, ennen kuin tiesi mistä aloittaa. Kirjastoista ei juuri löytynyt aiheeseen liittyviä kirjoja, joten lähteinä olivat e-kirjat ja www-sivut. Kaikki lähteet olivat englannin kielellä. HTTP-palvelinohjelmia olisi voinut optimoida enemmänkin, mutta se olisi vaatinut paljon enemmän työtä ja aikaa. Pyrin tekemään palvelinohjelmille ne optimoinnit, jotka selkeästi vaikuttivat suorituskykyyn. Testit eivät ole täysin luotettavia, koska testiohjelma ei vastaa oikeiden käyttäjien liikennöintiä, mutta ne antavat hyvän kuvan palvelinohjelmien suorituskyvystä ja optimoinnin hyödyistä.

Asennukset ja testien tekeminen ei sujunut ongelmitta, koska minulla ei ollut paljoa kokemusta komentorivipohjaisesta Linux käyttöjärjestelmästä. Käyttämällä Googlea ongelmat sai kuitenkin ratkottua. Aluksi harjoittelin virtuaalikoneilla, jotka asensin Wmware Workstationiin. Testiohjelma Siege toimi hyvin testauksessa.

Opinnäytetyön tekemisessä opin käyttämään Linuxin komentoriviä paremmin, pystyttämään HTTP-palvelimen ja tekemään säätöjä niihin. Opin myös ymmärtämään paremmin miten HTTP-palvelimet toimivat. Oppiminen tapahtui lukemalla lähteitä ja käytännön harjoituksilla ottamalla etäyhteys Raspberry Pi tietokoneeseen.

Jatkokehittämisideoita voisi olla muiden HTTP-palvelinohjelmien ottaminen mukaan vertailuun esim. Lighthttpd, Monkey tai Microsoftin ISS. HTTP-palvelimien tietoturva voisi olla myös jatkotutkimusta.

8 LÄHTEET

Smith, P. 2012. Professional Website Performance: Optimizing the Front-End and Back-End. Hakupäivä 14.10.2013,

http://proquest.safaribooksonline.com/book/web-development/9781118551721

Golstein, N. 2012. iOS Cloud Development For Dummies. Hakupäivä 16.10.2013, http://proquest.safaribooksonline.com/book/operating-systems-and-server-administration/virtualization/9781118235829

McManus, S & Cook, M. 2013. Raspberry Pi For Dummies. Hakupäivä 6.10.2013, http://proquest.safaribooksonline.com.ezp.oamk.fi:2048/book/hardware-and-gadgets/9781118554234

Webopedia. 2013. Web Server. Hakupäivä 13.11.2013,

http://www.webopedia.com/TERM/W/Web_server.html

Netcraft. 2013. Web Server Survey. Hakupäivä 4.10.2013,

http://news.netcraft.com/archives/2013/

Osoite. 2013. Kuormantasaus. Hakupäivä 7.12.2013,

http://www.osoite.fi/index.php/lisapalvelut/muut/kuormantasaus.html

Apache. 2013. FAQ. Hakupäivä

http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F

Slashroot. 2012. How is Nginx different from Apache. Hakupäivä 6.10.2013,

http://slashroot.in/how-is-nginx-different-from-apache

Nginx. 2013. Nginx. Hakupäivä 4.10.2013,

http://wiki.nginx.org/Main

Alexeev, A. 2013. Nginx. Hakupäivä 14.11.2013,

http://www.aosabook.org/en/nginx.html#fig.nginx.arch

Wordpress. 2013a. About. Hakupäivä 25.10.2013,

http://wordpress.org/about/

Wordpress. 2013b. Hosting. Hakupäivä 25.10.2013

http://wordpress.org/hosting/

Wordpress. 2013c. WP Super Cache. Hakupäivä 13.11.2013

http://wordpress.org/plugins/wp-super-cache/

Wisegeek. 2013. What Is a Web Server? Hakupäivä 12.10.2013, http://www.wisegeek.org/what-is-a-web-server.htm#

Lingan, J, B. 200. Web server. Hakupäivä 12.10.2013,

http://whatis.techtarget.com/definition/Web-server

Mertic, J. 2013. Five simple ways to tune your LAMP application. Hakupäivä 13.11-2013,

http://www.ibm.com/developerworks/library/os-5waystunelamp/

PHP. 2013. General Information. Hakupäivä 20.10.2013,

http://fi1.php.net/manual/en/faq.general.php

Fulmer, J. 2012. Siege-readme. Hakupäivä 7.10.2013,

http://www.joedog.org/siege-readme/

Rasbian. 2013. Welcome To Rasbian. Hakupäivä 6.10.2013,

http://www.raspbian.org/

Elinux. 2013. RPi Hardware. Hakupäivä 6.10.2013,

http://elinux.org/RPi_Hardware

Wikipedia 2013. GPIO. Hakupäivä 6.10.2013,

http://fi.wikipedia.org/wiki/GPIO

Raspberrypi 2013. FAQs. Hakupäivä 6.10.2013,

http://www.raspberrypi.org/faqs